



AF/2700  
Hull 5-4-1-4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application

Applicant(s): Hull et al.  
Case: 5-4-1-4  
Serial No.: 09/251,998  
Filing Date: February 19, 1999  
Group: 2143  
Examiner: David E. England

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature: *[Signature]* Date: August 13, 2003

Title: Eager Evaluation of Tasks in a Workflow System

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

RECEIVED

AUG 20 2003

Technology Center 2100

Sir:

Submitted herewith are the following documents relating to the above-identified patent application:

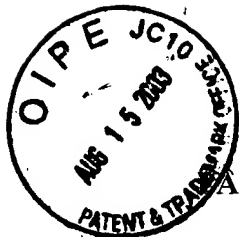
- (1) Appeal Brief (original and two copies); and
- (2) Copy of Notice of Appeal, filed on June 11, 2003, with copy of stamped return postcard indicating receipt of Notice by PTO on June 13, 2003.

There is an additional fee of \$320 due in conjunction with this submission under 37 CFR §1.17(c). Please charge **Deposit Account No. 50-0762** the amount of \$320, to cover this fee. In the event of non-payment or improper payment of a required fee, the Commissioner is authorized to charge or to credit **Deposit Account No. 50-0762** as required to correct the error. A duplicate copy of this letter and two copies of the Appeal Brief are enclosed.

Respectfully submitted,

Robert J. Mauri  
Attorney for Applicants  
Reg. No. 41,180  
Ryan, Mason & Lewis, LLP  
1300 Post Road, Suite 205  
Fairfield, CT 06824  
(203) 255-6560

Date: August 13, 2003



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Applicant(s): Hull et al.  
Case: 5-4-1-4  
Serial No.: 09/251,998  
Filing Date: February 19, 1999  
Group: 2143  
Examiner: David E. England

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

Signature: *[Signature]* Date: August 13, 2003

Title: Eager Evaluation of Tasks in a Workflow System

RECEIVED

AUG 20 2003

Technology Center 2100

APPEAL BRIEF

Mail Stop Appeal Brief-Patents  
Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

Applicants hereby appeal the final rejection dated March 20, 2003, of claims 1 through 21 of the above-identified patent application.

REAL PARTY IN INTEREST

The present application is assigned to Lucent Technologies Inc., as evidenced by an assignment recorded on May 21, 1999 in the United States Patent and Trademark Office at Reel 9970, Frame 0945. The assignee, Lucent Technologies Inc., is the real party in interest.

08/18/2003 HVUONG1 00000128 500762 09251998

01 FC:1402 320.00 DA

STATUS OF CLAIMS

The present application was filed on February 19, 1999, with claims 1-31. In a response to a restriction requirement, Applicants elected to prosecute claims 1-21. Consequently, claims 1-21 are currently pending. Claims 1 and 12 are independent claims. Claims 2-11 depend from independent claim 1, while claims 13-21 depend from independent claim 12. Independent claims 1 and 12 stand rejected under 102(e) as being anticipated by Hoenninger et al. (United States Patent No. 6,260,058, hereinafter

"Hoenninger"). Dependent claims 6 and 17 stand rejected under 35 U.S.C. §112 as being indefinite. Dependent claims 2-8 and 13-19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Codd et al. (United States Patent No. 6,421,667, hereinafter "Codd"). Dependent claim 9 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Lindsley (United States Patent No. 6,430,593). Dependent claims 10, 11, 20 and 21 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Van Praet et al. (United States Patent No. 5,854,929, hereinafter "Van Praet") in further view of Smith et al. (United States Patent No. 5,561,762, hereinafter "Smith").

#### STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

#### SUMMARY OF INVENTION

In a workflow system, an object is processed through execution of a number of tasks. An exemplary workflow system is shown in FIG. 1 of the drawings and is described on page 15, line 16 to page 9, line 18 of the specification. This workflow system is an object-focused workflow system that processes objects, which may be organized as modules. See, for example, page 6, lines 14-21. Modules have a number of enabling conditions associated with them. The enabling conditions indicate whether a module is to be executed for the object. FIG. 2 shows a ROUTING\_TO\_SKILL module having a number of other modules with associated enabling conditions. FIG. 2 is described on page 9, line 19 to page 13, line 8.

Tasks are associated with modules and are referred to by their associated modules. Tasks are described, e.g., on page 39, lines 11-17. Execution of one or more of the tasks results in initiation of a side-effect action performed by a component external to the workflow system. Side-effect actions are described, for instance, in reference to FIG. 4 and on page 13, line 21 to page 14, line 4. It is determined whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action. The task is executed using eager execution if the task is determined to be eligible for eager execution. States of tasks are

described, for instance, at page 40, line 20 to page 41, line 8 and page 64, lines 11-22. Algorithms for determining states of tasks and whether tasks should be executed eagerly are described in FIGS. 34A-34D and 35A-35D and associated text (e.g., page 41, line 9 to page 64, line 22).

#### ISSUES PRESENTED FOR REVIEW

- i. Whether independent claims 1 and 12 are properly rejected under 35 U.S.C. §102(e) as being anticipated by Hoenninger;
- ii. Whether dependent claims 6 and 17 are properly rejected under 35 U.S.C. §112 as being indefinite;
- iii. Whether dependent claims 2-8 and 13-19 are properly rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Codd;
- iv. Whether dependent claim 9 is properly rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Lindsley; and
- v. Whether dependent claims 10, 11, 20 and 21 are properly rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Van Praet in further view of Smith.

#### GROUPING OF CLAIMS

The rejected claims do not stand or fall together.

With regard to Issue (i), claims 1 and 12 stand or fall together.

With regard to Issue (ii), claims 6 and 17 stand or fall together.

With regard to Issue (iii), claims 2 and 13 stand or fall together, claims 3 and 14 stand or fall together, claims 6 and 17 stand or fall together, claims 8 and 19 stand or fall together, and claims 5, 7, 15 and 18 stand or fall together.

With regard to Issue (iv), claim 9 stand or falls alone.

With regard to Issue (v), claims 10 and 20 stand or fall together and claims 11 and 21 stand or fall together.

## ARGUMENT

### Issue (i), Rejection of Independent Claims 1 and 12

As to Issue (i) presented above, the Examiner rejected independent claims 1 and 12 under 35 U.S.C. §102(e) as being unpatentable over Hoenninger. The Examiner asserted that Hoenninger teaches all limitations of independent claims 1 and 12.

Applicants respectfully submit that Hoenninger does not teach or imply all limitations of amended independent claims 1 and 12. Amended independent claims 1 and 12 describe a method used by a workflow system and a workflow system, respectively. Both independent claims contain the limitation of “determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action.” This limitation will be called “the disputed limitation” herein. It should be noted, as claimed in the preambles to independent claims 1 and 12, that a side-effect action is “performed by a component external to said workflow system.” Applicants respectfully submit that Hoenninger does not teach or imply the disputed limitation and its two sub-limitations. In particular, no entity disclosed or implied in Hoenninger performs both sub-limitation (1) and sub-limitation (2). Applicants will describe two entities disclosed in Hoenninger: a system that performs the invention of Hoenninger; and a software developer that creates software defining tasks in Hoenninger.

As to the system disclosed in Hoenninger, Applicants read Hoenninger as allowing a complex control program to be divided into tasks, which are assigned priorities and activation events. See Abstract of Hoenninger. The Examiner points to certain sections of Hoenninger for the asserted disclosure of the limitation of independent claims 1 and 12. Applicants will describe certain of these sections in the argument that follows.

Applicants read Hoenninger as disclosing a system having an operating system and a “complex control program.” The complex control program is divided into a number of tasks, where the tasks have priorities and “activation events” associated therewith. See Abstract of Hoenninger. There is an “interrupt service routine” associated with a time counter in Hoenninger that determines which task should be activated at a

particular time. In order to activate a task, the interrupt service routine calls an operating system service responsible for task activation. See col. 9, lines 56-62 and also col. 10, lines 14-19 of Hoenninger. Each task has a state associated with it, as described in reference to FIGS. 3 and 4 of Hoenninger.

Even assuming, for sake of argument, that a state of a task in Hoenninger is used to determine whether a task is eligible for eager execution, Applicants can find no disclosure or implication in Hoenninger that the system in Hoenninger considers “whether execution of the task results in the initiation of a side-effect action” in order to determine whether a task is eligible for eager execution.

Hoenninger does state that “[e]ach task is then also characterized by an activation event that causes the task to be called up.” See col. 5, lines 63-65 of Hoenninger. Claim 4 of Hoenninger describes an activation event that “includes . . . the occurrence of external events.” But Hoenninger defines an activation event as causing a task to be called up (i.e., activated). In Applicants’ invention, a task can initiate a side-effect action, as shown in FIG. 4 of Applicants’ specification, for instance. In other words, in Hoenninger, an external event can cause a task to be activated, whereas in independent claims 1 and 12 of the present invention, a task initiates a side-effect action performed by a component external to the workflow system.

The Examiner asserts that the system disclosed in Hoenninger may be used to control an action outside of the controller 10 of Hoenninger. Even if this is true, for sake of argument, Applicants can find no disclosure or implication that the system in Hoenninger ever takes into account that a task may initiate a side-effect action performed by a component external to the workflow system when (or if) the system determines whether a task is to be eagerly executed, as set forth in independent claims 1 and 12.

Applicants respectfully submit that the system disclosed in Hoenninger never determines whether a task is eligible for eager execution by considering “whether execution of the task results in the initiation of a side-effect action,” as claimed by Applicants in independent claims 1 and 12.

With regard to a software developer that programs tasks in Hoenninger, the Examiner asserts that col. 2, lines 32-58 discloses the disputed limitation. Lines 40-58 of col. 2 of Hoenninger are as follows (emphasis added):

It is also advantageous that the tasks consist of a series of subtasks put together according to sequencing criteria. Sequencing criteria taken into account include the reason for the task processing request (activation event), the associated urgency (priority) and synchronization conditions between the subtasks. The configuration of many different subtasks according to sequencing criteria in processing sequences that are assigned to a few tasks with the respective priorities and reason for processing also reduces the running time required for sequencing control within the operating system which must be coordinated by processing the competing subtasks. *A significant amount of the information on sequencing control is thus already made available in writing the program and need not be determined while the program is running*, so that it adds to the running time. However, the complex controller program is divided into subtasks according to functional criteria. This increases transparency and simplifies the writing and management of programs.

Applicants respectfully submit that the sequencing control being described in the cited text is, as shown by emphasis, created BEFORE the program executes and the system never determines the cited sequencing control. In other words, a software developer performs the operations associated with the sequencing control, such as determining sequencing criteria. This is further shown by Hoenninger at col. 8, lines 44-46, where it states, “[t]he division of individual tasks A, B, C and D into a plurality of subtasks is performed by the *software developer*” (emphasis added).

As described above, the term “activation event” may be construed, as shown in claim 4 of Hoenninger, as “including . . . the occurrence of external events.” Nonetheless, as also described above, the entity performing the operations associated with the sequencing control, such as determining sequencing criteria, is a software developer. Furthermore, Hoenninger states, with regard to an action event, that: “[e]ach task is then also characterized by an activation event that causes the task to be called up.” Thus, Hoenninger explicitly describes an action event as causing a task to be called up, and not determining whether execution of a task results in initiation of a side-effect action. Even if the cited text of Hoenninger could, for sake of argument, be construed as “determining whether a task is eligible for eager execution by considering . . . whether execution of the task results in the initiation of a side-effect action,” the software developer never considers “a state of the task,” because such a state does not exist until the system operates. Additionally, the cited text specifically states that the cited types of sequencing control are NOT performed while the system operates.

Consequently, because the system of Hoenninger does not disclose an entity that performs both sub-limitations (1) and (2) of the disputed limitation in independent claims 1 and 12 of “determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action,” Applicants respectfully submit that independent claims 1 and 12 are not anticipated by Hoenninger.

Issue (ii), Rejection of Dependent Claims 6 and 17

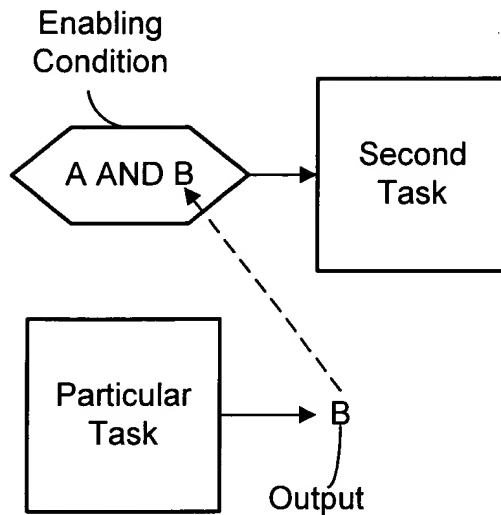
As to Issue (ii) presented above, the Examiner rejected claims 6 and 17 under 35 U.S.C. §112 as being indefinite for failing to point out and distinctly claim the subject matter which Applicants regard as their invention. In particular, the Examiner asserted that the limitation of these claims appear to be contradictory. Applicants respectfully disagree. The disputed language is as follows: “determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task’s enabling condition depends on one or more outputs of said particular task.”

Applicants explain in detail how unneeded tasks can be determined. See, for instance, FIGS. 34A-34D and associated text of Applicants’ specification. As part of this description, dependent claims 6 and 17 can correspond, for instance, to an example related on page 52, lines 7-11 of Applicants’ specification (claim language is emphasized below):

As a particular example, if the attribute [*particular task*] is an input for some target attribute [*second task*] but partial evaluation of the enabling condition [*e.g., “A AND B”*] for the target attribute [*second task*] indicates that the enabling condition [*e.g., “A AND B”*] will take the value FALSE [*i.e., because A is false*], and the attribute [*particular task*] will not be used, directly or indirectly in the evaluation of any other target attribute, then the node of the attribute [*particular task*] will become hidden.



This language may be described via the following diagram:



In this exemplary diagram, the enabling condition is “A AND B” and corresponds to the second task. The particular task produces an output of “B.” The dashed line between the output of the particular task and the enabling condition occurs because these relationships are generally implicit in the present invention. If it is known that “A” is false, then there is no need to determine B (i.e., because the enabling condition “A AND B” will be false if A is false), and the particular task need not be executed.

Consequently, Applicants respectfully submit that when claims 6 and 17 are read in light of Applicants’ specification, the claims are definite.

#### Issue (iii), Rejection of Dependent Claims 2-8 and 13-19

With regard to Issue (iii), claims 2 and 13 stand or fall together, claims 3 and 14 stand or fall together, claims 6 and 17 stand or fall together, claims 8 and 19 stand or fall together, and claims 5, 7, 15 and 18 stand or fall together. Applicants respectfully reiterate and incorporate by reference all arguments regarding Issue (i) for dependent claims 2-8 and 13-19. Therefore, these claims are believed allowable for at least the reasons identified above with respect to the independent claims.

The Examiner points to Hoenninger in view of Codd to render dependent claims 2-8 and 13-19 obvious. The Examiner asserts that Hoenninger does not disclose the limitations of dependent claims 2-8 and 13-19 but that Codd does, and the Examiner cites col. 16, lines 36-65 and 5 and col. 26, lines 22-40 of Codd as support for this

assertion. However, Applicants read col. 16, lines 36-65 of Codd as disclosing that tasks are executed when a truth value is TRUE. Applicants read col. 26, lines 22-40 of Codd as disclosing that a record can be created in a table and that the new table record can be processed. Evaluator output, based on the new table record, can lead to execution of a task correlator and task initiator.

With regard to claims 2 and 13, each of these claims contains a limitation relating to determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

Applicants can find no disclosure or implication in the cited text of Codd that the recited limitation in claims 2 and 13. In particular, the cited text of Codd does not describe side-effect actions and determining whether such side-effect actions are eligible for eager execution, and, as described above, neither does Hoenninger.

Similarly, claims 3 and 14 contain the limitation of “determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.” Applicants can find no disclosure or implication in the cited text of Codd or in Hoenninger of the recited limitation in claims 3 and 14. Again, determining whether side-effect actions are to be eagerly executed is not disclosed in Codd or Hoenninger.

As another example, dependent claims 6 and 17, described above, contain the limitation of “determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task’s enabling condition depends on one or more outputs of said particular task.” Applicants can find no disclosure or implication in the cited text of Codd that tasks are determined to be unneeded based on partial evaluation of enabling conditions, as claimed by dependent claims 6 and 17. In Codd, any “enabling conditions” have to be completely evaluated, and the cited text of Codd does not describe

determining unneeded tasks. Applicants can find no disclosure in Hoenninger of determining that tasks are unneeded for processing of an object.

Dependent claims 8 and 19 contain the limitation of “determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks’ enabling conditions depend on one or more outputs of said particular task.” Applicants can find no disclosure or implication in the cited text of Codd that a determination is made as to whether a particular task is necessary for processing of an object based on evaluation of enabling conditions for a number of tasks, as claimed in dependent claims 8 and 19. Again, Applicants respectfully submit that the cited text of Codd does not describe determining that tasks are necessary for processing of objects based on evaluation of enabling conditions.

Claims 5, 7, 15 and 18 are believed allowable for at least the reasons identified above with respect to the independent claims.

Issue (iv), Rejection of Dependent Claim 9

The Examiner rejected dependent claim 9 under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Lindsley. Applicants respectfully reiterate and incorporate by reference all arguments regarding Issue (i) for dependent claim 9. Therefore, this claim is believed allowable for at least the reasons identified above with respect to the independent claims.

Issue (v), Rejection of Dependent Claims 10, 11, 20 and 21

The Examiner rejected dependent claims 10, 11, 20 and 21 under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Van Praet in further view of Smith. The Examiner combined Van Praet and Hoenninger to reject claims 10 and 20, and Van Praet, Smith and Hoenninger to reject claims 11 and 21. Applicants respectfully reiterate and incorporate by reference all arguments regarding Issue (i) for dependent claims 10, 11, 20, and 21. Therefore, these claims are believed allowable for at least the reasons identified above with respect to the independent claims.

Additionally, claims 10 and 20 each contains limitations related to “wherein a memory of said workflow system stores a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions,” and “propagating changes through said graph based on new outputs of completed tasks.” Hoenninger does not disclose or imply these limitations, and Applicants read the cited text (col. 22, lines 7-14 and col. 8, lines 49-64) of Van Praet as providing formats used by a retargetable compiler, which then performs code generation. Data flow dependencies and enabling flow dependencies between tasks and enabling conditions are not described.

The remaining rejected dependent claims (i.e., claims 11 and 21) are believed allowable for at least the reasons identified above with respect to the independent claims.

Respectfully,



Date: August 13, 2003

Robert J. Mauri  
Attorney for Applicant(s)  
Reg. No. 41,180  
Ryan, Mason & Lewis, LLP  
1300 Post Road, Suite 205  
Fairfield, CT 06430  
(203) 255-6560

## APPENDIX

1. A method for operation of a workflow system for processing an object by executing a plurality of tasks, one or more of said tasks each having one or more associated enabling conditions indicating whether the task is to be executed for said object, and wherein execution of at least one of said tasks results in initiation of a side-effect action performed by a component external to said workflow system, said method comprising the steps of:

determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action; and

executing the task using eager execution if the task is determined to be eligible for eager execution.

2. The method of claim 1 wherein the step of determining whether a task is eligible for eager execution further comprises the step of:

determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

3. The method of claim 1 wherein the step of determining whether a task is eligible for eager execution further comprises the step of:

determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

4. The method of claim 1 wherein said step of determining whether a task is eligible for eager execution further comprises the step of:

partially evaluating one or more enabling conditions associated with said task.

5. The method of claim 1 wherein said step of determining whether a task is eligible for eager execution is performed by also considering (3) whether the task contributes to the production of a target value.

6. The method of claim 1 further comprising the step of:

determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task.

7. The method of claim 1 further comprising the step of:

determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on said particular task.

8. The method of claim 1 further comprising the step of:

determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on one or more outputs of said particular task.

9. The method of claim 1 wherein said step of determining is performed repeatedly during the processing of the object.

10. The method of claim 1 wherein a memory of said workflow system stores a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions, said method further comprising the step of:

propagating changes through said graph based on new outputs of completed tasks.

11. The method of claim 10 wherein said step of propagating changes is based on predefined propagation rules.

12. A workflow system for processing an object by executing a plurality of tasks, one or more of said tasks each having one or more associated enabling conditions

indicating whether the task is to be executed for said the object, and wherein execution of at least one of said tasks results in initiation of a side-effect action performed by a component external to said workflow system, said system comprising:

means for determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action; and

means for executing the task using eager execution if the task is determined to be eligible for eager execution.

13. The workflow system of claim 12 wherein the means for determining whether a task is eligible for eager execution further comprises:

means for determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

14. The workflow system of claim 12 wherein the means for determining whether a task is eligible for eager execution further comprises:

means for determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.



15. The workflow system of claim 12 wherein said means for determining whether a task is eligible for eager execution further comprises:

means for partially evaluating one or more enabling conditions associated with said task.

16. The workflow system of claim 12 wherein said means for determining whether a task is eligible for eager execution further comprises:

means for determining whether the task contributes to the production of a target value.

17. The workflow system of claim 12 further comprising:

means for determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task.

18. The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on said particular task.

19. The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on one or more outputs of said particular task.

20. The workflow system of claim 12 further comprising:

a memory for storing a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions; and

means for propagating changes through said graph based on new outputs of completed tasks.

21. The workflow system of claim 20 wherein said memory stores predefined propagation rules and wherein said means for propagating changes further comprises means for propagating changes based on said predefined propagation rules.